**IN THE CLAIMS:**

Please amend the claims as follows:

1. (currently amended) A method of optimizing scalability in a multiprocessor data server having N processors, wherein N is an integer greater than or equal to 2, the method comprising:

implementing N network interface cards (NICs) ~~(Network Interface Cards)~~, a first one of the N NICs being dedicated to receiving an incoming data stream;

binding an interrupt from the first one of the N NICs to a first one of the N processors;

binding an interrupt for an nth NIC to an nth processor, wherein 0 < n < = N; and

binding a deferred procedure call (DPC) ~~(Deferred Procedure Call)~~ for the nth NIC to the nth processor.

2. (original) The method of claim 1, further comprising tightly coupling M client connections to the nth processor via the nth NIC, wherein M is a positive integer.

3. (currently amended) The method of claim 1, further comprising binding P server threads to specific ones of the second through Nth processors, wherein P is a positive integer.

4. (currently amended) The method of claim 2, further comprising binding P server threads to specific ones of the second through Nth processors, wherein P is a positive integer.

5. (currently amended) The method of claim 1, further comprising:

defining ~~L1 (Level 1) and L2 (Level 2)~~ first and second level caches for each of the N processors;

storing instructions and temporal data in ~~L2~~ second level caches of the N processors; and

storing non-temporal data in ~~L1~~ first level caches of the N processors, bypassing the ~~L2~~ second level caches.

6. (currently amended) The method of claim 2, further comprising:

defining ~~L1 (Level 1) and L2 (Level 2)~~ first and second level caches for each of the N processors;

storing instructions and temporal data in ~~L2~~ second level caches of the N processors; and

storing non-temporal data in ~~L1~~ first level caches of the N processors, bypassing the ~~L2~~ second level caches.

7. (currently amended) The method of claim 3, further comprising:

defining ~~L1 (Level 1) and L2 (Level 2)~~ first and second level caches for each of the N processors;

storing instructions and temporal data in ~~L2~~ second level caches of the N processors; and

storing non-temporal data in ~~L1~~ first level caches of the N processors, bypassing the ~~L2~~ second level caches.

8. (currently amended) The method of claim 4, further comprising:

defining ~~L1 (Level 1) and L2 (Level 2)~~ first and second level caches for each of the N processors;

storing instructions and temporal data in ~~L2~~ second level caches of the N processors; and

storing non-temporal data in ~~L1~~ first level caches of the N processors, bypassing the ~~L2~~ second level caches.

9. (currently amended) A method of ~~optimizing~~ providing scalability in a multiprocessor data server having N processors, wherein N is an integer greater than or equal to 2, the method comprising:

implementing N network interface cards (NICs) ~~(Network Interface Cards)~~; and

tightly coupling M client connections to the nth processor via the nth NIC, wherein M is a positive integer and wherein 0 < n < = N.

10. (currently amended) The method of claim 9, further comprising binding P server threads to specific ones of the second through Nth processors, wherein P is a positive integer.

11. (currently amended) The method of claim 10, further comprising:

defining ~~L1 (Level 1) and L2 (Level 2)~~ first and second level caches for each of the N processors;

storing instructions and temporal data in ~~L2~~ second level caches of the N processors; and

storing non-temporal data in ~~L1~~ first level caches of the N processors, bypassing the ~~L2~~ second level caches.

12. (currently amended) The method of claim 9, further comprising:

defining ~~L1 (Level 1) and L2 (Level 2)~~ first and second level caches for each of the N processors;

storing instructions and temporal data in ~~L2~~ second level caches of the N processors; and

storing non-temporal data in ~~L1~~ first level caches of the N processors, bypassing the ~~L2~~ second level caches.

13. (currently amended) A method of ~~optimizing~~ providing scalability in a multiprocessor data server having N processors, wherein N is an integer greater than or equal to 2, the method comprising:

implementing N network interface cards (NICs) ~~(Network Interface Cards)~~; and

binding P server threads to specific ones of ~~a~~ the second through Nth processors.

14. (currently amended) The method of claim 13, further comprising:

defining ~~L1 (Level 1) and L2 (Level 2)~~ first and second level caches for each of the N processors;

storing instructions and temporal data in ~~L2~~ second level caches of the N processors; and

storing non-temporal data in ~~L1~~ first level caches of the N processors, bypassing the ~~L2~~ second level caches.

15. (currently amended) A method of ~~optimizing~~ providing scalability in a multiprocessor data server having N processors, wherein N is an integer greater than or equal to 2, the method comprising:

implementing ~~L1 (Level 1) and L2 (Level 2)~~ first and second level caches for each of the N processors;

storing instructions and temporal data in ~~L2~~ second level caches of the N processors; and

storing non-temporal data in ~~L1~~ first level caches of the N processors, bypassing the ~~L2~~ second level caches.

16. (currently amended) The method of claim 5, further comprising improving ~~L1~~ first level cache efficiency by increasing a time quantum allotted to server threads which process streaming data buffers.

17. (currently amended) The method of claim 6, further comprising improving ~~L1~~ first level cache efficiency by increasing a time quantum allotted to server threads which process streaming data buffers.

18. (currently amended) The method of claim 7, further comprising improving ~~L1~~ first level cache efficiency by increasing a time quantum allotted to server threads which process streaming data buffers.

19. (currently amended) The method of claim 8, further comprising improving ~~L1~~ first level cache efficiency by increasing a time quantum allotted to server threads which process streaming data buffers.

20. (currently amended) The method of claim 11, further comprising improving ~~L1~~ first level cache efficiency by increasing a time quantum allotted to server threads which process streaming data buffers.

21. (currently amended) The method of claim 14, further comprising improving ~~L1~~ first level cache efficiency by increasing a time quantum allotted to server threads which process streaming data buffers.

22. (currently amended) The method of claim 15, further comprising improving ~~L1~~ first level cache efficiency by increasing a time quantum allotted to server threads which process streaming data buffers.

23. (currently amended) A multiprocessor data server comprising:

N processors, wherein N is an integer greater than or equal to 2;

N network interface cards (NICs) ~~(Network Interface Cards)~~, a first one of said N NICs being dedicated to receiving an incoming data stream;

wherein an interrupt from the first one of said N NICs is bound to a first one of said N processors; and

wherein an interrupt for an nth NIC is bound to an nth processor, 0 < n < = N; and

wherein a deferred procedure call (DPC) ~~(Deferred Procedure Call)~~ for said nth NIC is bound to said nth processor.

24. (original) The multiprocessor data server of claim 23, further comprising M client connections, wherein said M client connections are tightly coupled to said nth processor via said nth NIC, M being a positive integer.

25. (currently amended) The multiprocessor data server of claim 23, further comprising P server threads, wherein said P server threads are bound to specific ones of ~~a~~ the second through Nth processors.

26. (currently amended) The multiprocessor data server of claim 24, further comprising P server threads, wherein said P server threads are bound to specific ones of ~~a~~ the second through Nth processors.

27. (currently amended) The multiprocessor data server of claim 23, further comprising ~~L1 (Level 1) and L2 (Level 2)~~ first and second level caches for each of said N processors, wherein instructions and temporal data are stored in said ~~L2~~ second level caches of said N processors, and wherein non-temporal data is stored in ~~L1~~ first level caches of said N processors, bypassing the ~~L2~~ second level caches.

28. (currently amended) The multiprocessor data server of claim 24, further comprising ~~L1 (Level 1) and L2 (Level 2)~~ first and second level caches for each of said N processors, wherein instructions and temporal data are stored in said ~~L2~~ second level caches of said N processors, and wherein non-temporal data is stored in ~~L1~~ first level caches caches of said N processors, bypassing the ~~L2~~ second level caches.

29. (currently amended) The multiprocessor data server of claim 25, further comprising ~~L1 (Level 1) and L2 (Level 2)~~ first and second level caches for each of said N processors, wherein instructions and temporal data are stored in said ~~L2~~ second level caches of said N processors, and wherein non-temporal data is stored in ~~L1~~ first level caches caches of said N processors, bypassing the ~~L2~~ second level caches.

30. (currently amended) The multiprocessor data server of claim 26, further comprising ~~L1 (Level 1) and L2 (Level 2)~~ first and second level caches for each of said N processors, wherein instructions and temporal data are stored in said ~~L2~~ second level caches of said N processors, and wherein non-temporal data is stored in ~~L1~~ first level caches caches of said N processors, bypassing the ~~L2~~ second level caches.

31. (currently amended) A program storage device, readable by a machine, embodying a program of instructions executable by the machine to perform a method of ~~optimizing~~ providing scalability in a multiprocessor data server having N processors, wherein N is an integer greater than or equal to 2, the method comprising:

implementing N network interface cards (NICs) ~~(Network Interface Cards)~~, a first one of the N NICs being dedicated to receiving an incoming data stream;

binding an interrupt from the first one of the N NICs to a first one of the N processors;

binding an interrupt for an nth NIC to an nth processor, wherein $0 < n < = N$; and

binding a deferred procedure call (DPC) ~~(Deferred Procedure Call)~~ for the nth NIC to the nth processor.

32. (original) The program storage device of claim 31, the method further comprising tightly coupling M client connections to the nth processor via the nth NIC, wherein M is a positive integer.

33. (currently amended) The program storage device of claim 31, the method further comprising binding P server threads to specific ones of the second through Nth processors, wherein P is a positive integer.

34. (currently amended) The program storage device of claim 32, the method further comprising binding P server threads to specific ones of the second through Nth processors, wherein P is a positive integer.

35. (currently amended) The program storage device of claim 31, the method further comprising:

defining ~~L1 (Level 1) and L2 (Level 2)~~ first and second level caches for each of the N processors;

storing instructions and temporal data in ~~L2~~ second level caches of the N processors; and

storing non-temporal data in ~~L1~~ first level caches of the N processors, bypassing the ~~L2~~ second level caches.

36. (currently amended) The program storage device of claim 32, the method further comprising:

defining ~~L1 (Level 1) and L2 (Level 2)~~ first and second level caches for each of the N processors;

storing instructions and temporal data in ~~L2~~ second level caches of the N processors; and

storing non-temporal data in ~~L1~~ first level caches of the N processors, bypassing the ~~L2~~ second level caches.

37. (currently amended) The program storage device of claim 33, the method further comprising:

defining ~~L1 (Level 1) and L2 (Level 2)~~ first and second level caches for each of the N processors;

storing instructions and temporal data in ~~L2~~ second level caches of the N processors; and

storing non-temporal data in ~~L1~~ first level caches of the N processors, bypassing the ~~L2~~ second level caches.

38. (currently amended) The program storage device of claim 34, the method further comprising:

defining ~~L1 (Level 1) and L2 (Level 2)~~ first and second level caches for each of the N processors;

storing instructions and temporal data in ~~L2~~ second level caches of the N processors; and

storing non-temporal data in ~~L1~~ first level caches of the N processors, bypassing the ~~L2~~ second level caches.